

State space:

Abstract:

Here we will introduce the huge topic of state spaces. It is relevant in classic AI, specifically reaching goal states from other states, and in RL and in control theory where we include also actions to transverse among states.

Then we will discuss the different search methods to guide different states to the goal states. Methods such as programming or control.

In ML we will visualize the state space before learning and after it, encouraging generalization to unseen data. The learning occurs in hypothesis space.

We'll see that in classical AI, we either search starting from the initial states or backward from the goal states, and sometimes even by their combination.

We'll see different possible representations, such as vector space, the function which assigns each state with a value, and a formal logic one.

In formal logic the state comprises of objects and their features, with action changing the state's inner structure.

In RL we will assume the Markov property for next state's fully dependence on previous state only.

Then we'll see the dynamics in RL either without intervention, that is full autonomy, or by observing hidden states via output, or with intervention, that is some input action.

We'll see how different elements in RL can be either deterministic or stochastic.

We'll see three main methods to learn optimal policies in RL.

DP by considering current state's reward and all next states.

MC by running random experiments via a given policy.

And TD that uses an error between predicted and current values, for the current and next states.

There are also other versions of TD, differing by how action is selected.

All of these methods use the evaluation function to evaluate how good is a policy, so that later to compute a better policies via other algorithms.

Summary:

Here we introduce the huge topic of state spaces. It is relevant in classic AI, specifically reaching goal states from other states, and in RL and in control theory where we include also actions to transverse among states.

We start by formulating state space for this purpose of search as derived from some will, which determines the actions to use to reach the goal states.

So we have some vector field representing this will, which influences the actions that are eventually selected.

Now we can analyze different search methods to guide different states to the goal states.

For example, in programming or control the user abstract its will, that is make it general, and manually calculates how to realize it for a huge set of possible inputs or initial states to reach some of the plausible goal states.

In ML we can visualize the data before applying any learning as colors for labels or as colors for clusters. After learning we are interesting to generalize from this data for any new incoming data, hence we visualize the state space as continuous partitioned regions, that assign any new data point with its label or cluster.

Also in ML, we concentrate mostly on hypothesis search, that is for the best sets of parameters or model to not only describe the given data but also unseen data, meaning again, generalize.

In classical AI, we either search starting from the initial states or backward from the goal states, and sometimes even by combining these two approaches. Also instead of a graph, we can represent this search via tree.

Alternatively, we can assign value for each state, thus convert the search to be on a manifold or a function, usually via gradient methods.

We see some of the possible representations, such as vector space, the function, and a formal logic one. In all of them, the user will is embedded in the state, and through some function selects the next action to perform. In formal logic the state comprises of objects and their features, with action changing the state's inner structure.

In RL we also assume an action space combined with states. Only we assume a Markov property holds. It is that every state encapsulate all the necessary information to decide about the next state, such that no previous states are needed.

We can see the dynamics in RL either without intervention, that is full autonomy, where the states are evolved by some function or some probability. This probability can be simplified and factorized using the Markov property.

In case the states are hidden or unobservable, we must have some measurements or outputs from the system. And in case of intervention, we have some action applied externally. This action could be deterministically derived from the given state, such as greedy or optimal policy, or stochastically, with probabilities for different possible actions.

We can view some illustration of these cases, with strong relevance to causality models.

There are three main methods to learn optimal policies in RL. We first assume some functions to evaluate given policies, either for a state or including also taken action.

DP uses bellman equation to improve the evaluation function, by considering current state's reward and all next states values of this function, thus it requires knowing of the transition model.

MC improve the evaluation function, by running random experiments via the given policy, and updating its values backward from the terminal state. It doesn't need transition model for that.

TD uses the error between predicted value comprised of the current reward and next state's value, and the current's state value to improve and update the evaluation function. It uses only the current and the actual next state.

Other versions of TD are SARSA with action selected randomly from a given policy, q-learning with actions selected greedily, i.e. optimally, and extended SARSA where the action is averaged.

All of these methods use the evaluation function to evaluate how good is a policy, thus to compute a better policies via other algorithms.

Also note that sometimes the learned policy is different from the one observed. This happens in MC, which requires the use of importance sampling to learn the correct evaluation for the learned policy, and in TD in q-learning and extended SARSA which learn different actions than the ones that are actually applied.

Finally, if we allow more rewards from states in TD, it becomes more and more resembling the MC method.